Amendments to the Claims:

1 (currently amended): A method for signing frame transmissions from a broadcast server to a client device, comprising:

~~retrieving~~ obtaining a data block that is scheduled for transmission in ~~the~~ a next frame; wherein the next frame includes a number (n) of data blocks;

selecting a secret key (S_n) that is associated with the client device for ~~a~~ a predetermined number ~~(n)~~ of the data blocks in the frame;

generating a count that is associated with a time;

computing a set of hash keys using the secret key (S_n) and ~~a~~ the count ~~that is associated with time;~~

selecting a hash key (S_i) that is associated with the data block, wherein the selected hash key corresponds to one of the set of hash keys;

computing ~~an~~ a keyed-hash message authentication code (HMAC) ~~HMAC~~ value for the next frame using the selected hash key (S_i);

periodically signing and transmitting a datum containing the hash key of an earlier or initial frame with a digital signature key (K_S) ; and

assembling the next frame such that the data block and the HMAC value appear before the hash key in the frame transmission.

2 (original): The method of claim 1 wherein the datum corresponds to at least one of $\{n, S_0)\}K_s$ and (n, b, S_b) where b corresponds to a preceding frame number from a previous frame transmission.

3 (original): The method of claim 1, further comprising: selecting the count such that the count is associated with an index of the data block.

4 (original): The method of claim 1, further comprising: selecting the count such that the count corresponds to a time stamp associated with an internal clock in the broadcast server.

5 (original): The method of claim 1, wherein computing the set of hash keys corresponds to applying a one-way hashing function to the secret key (S_n) for n iterations such that $S_i = \text{HASH}(S_{i+1})$.

6 (original): The method of claim 1, wherein computing the HMAC value corresponds to a hashed message authentication code, wherein a value (H_i) associated with the hashed message authentication code is given as $H_i = \text{HMAC}(F_i, S_i)$, where F_i corresponds to the data being signed, S_i the key for signing, and i the sequence number associated with the data and key.

7 (original): The method of claim 1, further comprising: selecting a new secret key as the secret key (S_n) when the previous secret key has been applied to n data blocks in the next frame.

8 (currently amended): The method of claim 1, wherein periodically signing the datum comprises ~~at least one of~~ signing the datum ~~for~~ every frame~~, and signing the datum over an interval that does not correspond to every frame~~.

9 (original): The method of claim 1, further comprising: incrementing the count before retrieving a data block that is scheduled for transmission in the next frame.

10 (currently amended): The method of claim 9, wherein incrementing the count corresponds to at least one of: incrementing a time ~~step~~ stamp in the broadcast server, incrementing the frame number associated with the next frame that is scheduled for transmission, and incrementing the block number associated with the next data block in the next frame that is scheduled for transmission.

11 (currently amended): A method for authenticating frame transmissions from a server to a client device, comprising:

retrieving an Rivest Shamir Adleman (RSA) ~~RSA~~ signed datum from a frame;

verifying an RSA signature associated with the RSA signed datum from the frame;

storing a hash key (S_0) that is associated with the frame when the RSA signature is verified;

retrieving another hash key (S_i) and an keyed-hash message authentication code (HMAC) ~~HMAC~~ value from the frame;

verifying the other hash key (S_i) that is obtained from a previous frame;

verifying the HMAC value with the other hash key (S_i);

discarding the frame when at least one of the other hash key (S*i*) and the HMAC value fail verification; and

accepting the frame when the other hash key (S_i) and the HMAC value are successfully verified.

12 (currently amended): The method of claim 11, further comprising evaluating a count associated with the client device, computing a hash key using the count and a secret key (S_n) that is known by both the server and the client device, wherein the count corresponds to at least one of: a time ~~step~~ stamp in the client device, identifying the frame number associated with the frame, and identifying the block number that is associated with the frame.

13 (original): The method of claim 11, wherein verifying the other hash key (S_i) comprises: retrieving a previously stored hash key, retrieving a count in the client device, computing an expected hash key from the previously stored hash key and the count, and comparing the expected hash key to the other hash key (Si).

14 (original): The method of claim 13, wherein the count corresponds to at least one of: a time step in the client device, identifying the frame number associated with the frame, and identifying the block number that is associated with the frame.

15 (original): The method of claim 11, wherein verifying the HMAC value with the other hash key (S_i) comprises: computing a value (H_i) that is associated with a hashed message authentication code as given by $H_i = HMAC(F_i, S_i)$, where F_i corresponds to the data being

signed, S_i the key for signing, and i the sequence number associated with the data and key, and comparing the computed value with the retrieved HMAC value from the frame.

16 (original): The method of claim 11, further comprising: storing a verified hash key (S_i) for verification of further transmission frames after the hash key is accepted.

17 (original): A broadcast communication system for communicating frame transmissions from a server to a client device, comprising:

a scheduler that is arranged to provide data blocks to the server for transmission in a next frame;

a counter that is arranged to provide a count in the server;

a hashing function in the server that is arranged to compute hash keys for the next frame using the count and a secret key;

an HMAC function in the server that is arranged to provide an HMAC value in response to hash keys associated with the next frame;

a broadcast processor in the server that is arranged to receive the hash keys, HMAC values, and the data blocks, and organize the next frame for transmission

such that the data block and the HMAC value appear before the hash key in the frame transmission.

18 (original): The broadcast communication system of claim 17, further comprising:

a broadcast receiver in the client device that is arranged to receive a transmitted frame, wherein the transmitted frame starts with another HMAC value, continues with another signed datum$\{n, S_0)\}K_s$ followed by another data block, and ends with another hash key S_i;

a counter in the client device that is arranged to provide another count;

a hashing function in the client device that is arranged to compute additional hash keys for the frame transmission using the other count, the secret key, and previously stored hash keys;

a verification function block in the client device that is arranged to verify the other hash key (S_i) with the additional hash keys and verify the HMAC value with the other hash key (S_i) and previous hash keys;

a means for discarding the frame in the client device when at least one of the other hash key (S_i) and the HMAC value fail verification; and

a means for accepting the frame in the client device when the other hash key (S_i) and the HMAC value are successfully verified.

19 (original): The broadcast communication system of claim 18, further comprising: a means for recording the other hash key (S_i) when the frame is accepted, wherein the other hash key (S_i) is utilized for verification of subsequently received transmission frames.

20 (original): A system for authenticating frame transmissions in a client device, comprising:

a broadcast receiver that is arranged to receive a transmitted frame, wherein the transmitted frame includes an HMAC value and a data block, and ends with a hash key S_i;

a counter that is arranged to provide a count that has a time dependence;

a hashing function that is arranged to compute hash keys for the transmitted frame using the count and a secret key;

a verification function block that is arranged to verify the hash key (S_i) with the computed hash keys, and also arranged to verify the HMAC value with the hash key (S_i) and the previously stored hash keys;

a means for discarding the frame when at least one of the hash key (S_i) and the HMAC value fail verification;

a means for accepting the frame when the hash key (S_i) and the HMAC value are successfully verified; and

a means for storing the hash key as a previously stored hash key when the frame is accepted such that subsequent frames utilize the stored hash key for verification.